HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

04-21-06

PATENT APPLICATION

ATTORNEY DOCKET NO. 10017028-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	George S. Gales et al.	Confirmation No.:	3057
Application No.:	10/001,410	Examiner:	Arezoo Sherkat
Filing Date:	10/31/2001	Group Art Unit:	2131

Title: SYSTEM AND METHOD OF DEFINING THE SECURITY VULNERABILITIES OF A COMPUTER SYSTEM

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on February 22, 2006.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500 . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV568259959US addressed to: MS Appeall Brief – Patents Commissioner for Patents, Alexandria, VA 22313-1450 Date of Deposit: April 20, 2006

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile:

Typed Name: Elise Perkins

Signature: Elise Perkins

Respectfully submitted,

George S. Gales et al.

By _____

Jody C. Bishop

Attorney/Agent for Applicant(s)

Reg No. : 44,034

Date : April 20, 2006

Telephone : (214) 855-8007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 10017028-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
George S. Gales et al.

Application No.: 10/001,410 | Confirmation No.: 3057

Filed: October 31, 2001 | Art Unit: 2131

For: SYSTEM AND METHOD OF DEFINING THE SECURITY VULNERABILITIES OF A COMPUTER SYSTEM | Examiner: Arezoo Sherkat

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on February 22, 2006, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I. Real Party In Interest
II Related Appeals and Interferences
III. Status of Claims
IV. Status of Amendments
V. Summary of Claimed Subject Matter
VI. Grounds of Rejection to be Reviewed on Appeal
VII. Argument
VIII. Claims Appendix

IX. Evidence Appendix
X. Related Proceedings Appendix

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Limited Partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 27 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-27
4. Claims allowed: None
5. Claims rejected: 1-27

C. Claims On Appeal

The claims on appeal are claims 1-27

IV. STATUS OF AMENDMENTS

A Final Office Action rejecting the claims of the present application was mailed December 22, 2005. In response, Applicant did not file an Amendment After Final Rejection, but instead filed a Notice of Appeal, which this brief supports. Accordingly, the claims on appeal are those as rejected in the Final Office Action of December 22, 2005. A complete listing of the claims is provided in the Claims Appendix hereto.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. It should be noted that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

According to one claimed embodiment, such as that of independent claim 1, a method of defining the security vulnerability of a computer system comprises generating a human-readable and machine-readable vulnerability description language (VDL) file (e.g., VDL file 200 of FIG. 3, and *see* page 8, line 14 – page 9, line 12 and page 11, lines 10-19 of the specification). The VDL file specifies: an attack representing a recognized vulnerability of the computer system (*see* page 3, lines 21-24, page 9, lines 26-28, and page 22, lines 1-16 of the specification); at least one attribute of the specified attack (*see* page 3, line 24, page 9, line 26 – page 10, line 9, and page 22, lines 1-16 of the specification); at least one policy definition with respect to detecting the vulnerability of the computer system to the specified attack (*see* page 3, lines 24-26, page 10, lines 23-27, page 13, lines 25-31, page 22, lines 1-10, and page 22, lines 20-25 of the specification); and a remedy for the specified vulnerability (*see* page 3, line 27, and page 9, lines 26-33 of the specification).

In certain embodiments, such as that of claim 3, the method further comprises generating the VDL file specifying a computing platform of the computer system (*see* page 4, line 2, page 8, lines 11-13, and page 9, line 26 – page 10, line 9 of the specification).

In certain embodiments, such as that of claim 4, the method further comprises generating the VDL file: specifying a security category of the specified attack (*see* page 11, line 19 – page 12, line 20, and page 13, lines 16-24 of the specification); and specifying at least one policy group with respect to the specified security category (*see* page 11, line 19 – page 12, line 20, and page 12, line 36 – page 13, line 15 of the specification).

In certain embodiments, such as that of claim 5, the method further comprises generating the VDL file specifying a vulnerability scanner executing on the computer system (e.g., block 208 of FIG. 3, and *see* page 12, lines 21-25 of the specification).

In another claimed embodiment, such as that of independent claim 12, a method of defining a security vulnerability condition of a system comprises generating a human-readable and machine-readable vulnerability description language (VDL) file (e.g., VDL file 200 of FIG. 3, and *see* page 8, line 14 – page 9, line 12 and page 11, lines 10-19 of the specification). The VDL file specifies: a name of a vulnerability associated with the system (*see* page 3, lines 30-31, page 9, lines 26-28, page 11, line 19 – page 12, line 20, and page 22, lines 1-16 of the specification); at least one attribute of the specified vulnerability (*see* page 3, lines 31-32, page 9, line 26 – page 10, line 9, and page 22, lines 1-16 of the specification); a remedy for the vulnerability according to the specified computing platform (*see* page 4, lines 2-3, and page 9, line 26 – page 10, line 9 of the specification); a policy definition with respect to detecting the specified vulnerability (*see* page 3, line 32 – page 4, line 1, page 10, lines 23-27, page 13, lines 25-31, page 22, lines 1-10, and page 22, lines 20-25 of the specification); and at least one attribute of the specified policy definition (*see* page 4, line 1, page 10, lines 23-27, page 13, lines 25-31, page 22, lines 1-10, and page 22, lines 20-25 of the specification).

In certain embodiments, such as that of claim 13, the method further comprises specifying a computing platform of the system (*see* page 4, line 2, page 8, lines 11-13, and page 9, line 26 – page 10, line 9 of the specification).

In certain embodiments, such as that of claim 14, the method further comprises specifying a security category of the specified vulnerability (*see* page 11, line 19 – page 12, line 20, and page 13, lines 16-24 of the specification); and specifying at least one policy

group with respect to the specified security category (*see* page 11, line 19 – page 12, line 20, and page 12, line 36 – page 13, line 15 of the specification).

In certain embodiments, such as that of claim 15, the method further comprises specifying a vulnerability scanner executing on the system (e.g., block 208 of FIG. 3, and *see* page 12, lines 21-25 of the specification).

In another claimed embodiment, such as that of independent claim 20, a system of defining security vulnerabilities of a computer system comprises a human-readable and machine-readable vulnerability description language (VDL) file (e.g., VDL file 200 of FIG. 3, and *see* page 8, line 14 – page 9, line 12 and page 11, lines 10-19 of the specification). The VDL file contains a definition of at least one vulnerability (*see* page 4, line 6, page 9, lines 26-28, and page 22, lines 1-16 of the specification) and a definition of at least one policy item for detecting the vulnerability (*see* page 4, lines 6-7, page 10, lines 23-27, page 13, lines 25-31, page 22, lines 1-10, and page 22, lines 20-25 of the specification). The system further comprises an interpreter operable to parse the at least one vulnerability definition and at least one policy item definition in the VDL file and organize the parsed definitions pursuant to a predetermined format (e.g., interpreter 202 of FIG. 3, and *see* page 4, lines 7-10, and page 9, lines 13-25 of the specification). The system further comprises a data storage operable to store the parsed and organized at least one vulnerability and at least one policy item definition (e.g., configuration database 204 of FIG. 3, and *see* page 4, lines 10-12, and page 9, lines 13-25 of the specification), wherein the data storage is accessible by at least one vulnerability scanner application (e.g., vulnerability assessment application 208 of FIG. 3).

In certain embodiments, such as that of claim 22, the VDL file further comprises a definition of a vulnerability scanner application (*see* page 12, lines 21-25 of the specification).

In certain embodiments, such as that of claim 23, the VDL file further comprises a definition of a security category providing a grouping of the at least one vulnerability (*see* page 11, line 19 – page 12, line 20, and page 13, lines 16-24 of the specification), and a definition of a policy group providing a grouping of the at least one policy item (*see* page 11, line 19 – page 12, line 20, and page 12, line 36 – page 13, line 15 of the specification).

VI. GROUNDS OF OBJECTION TO BE REVIEWED ON APPEAL

Claims 1-27 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application Publication No. 2002/0116639 to Chefalas et al. (hereinafter "*Chefalas*") in view of U.S. Patent No. 6,088,804 to Hill et al. (hereinafter "*Hill*").

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. Instead, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

A. Rejections under 35 U.S.C. §103(a) over *Chefalas* in view of *Hill*

Claims 1-27 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Chefalas* in view of *Hill*. Appellant respectfully traverses this rejection below.

To establish a prima facie case of obviousness, three basic criteria must be met. *See* M.P.E.P. § 2143. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the applied references must teach or suggest all the claim limitations. Without conceding any other criteria, the applied references fail to teach or suggest all elements of the claims, as discussed hereafter.

Independent Claim 1 and Dependent Claims 2 and 6-11

Independent claim 1 recites:

> A method of defining the security vulnerability of a computer system, comprising:
> generating a human-readable and machine-readable vulnerability description language (VDL) file specifying:
> an attack representing a recognized vulnerability of the computer

system;
at least one attribute of the specified attack;
at least one policy definition with respect to detecting the vulnerability of the computer system to the specified attack; and
a remedy for the specified vulnerability.

The combination of *Chefalas* and *Hill* fails to teach or suggest generating such a VDL file, as discussed hereafter. It is unclear what teaching of the references the Final Office Action relies upon as providing the VDL file. The Final Office Action appears to assert on page 2 thereof that *Chefalas* teaches generating a human-readable and machine-readable VDL file. For instance, the Final Office Action asserts that "Chefalas discloses a method of defining the security vulnerability of a computer system, comprising: generating a human-readable and a machine-readable vulnerability description language (VDL) file (i.e., Network Status Display 42)...." Thus, the Final Office Action appears to rely upon *Chefalas'* teaching of a Network Status Display 42 as providing the recited VDL file. However, *Chefalas* makes no mention whatsoever of a network status display. Thus, it is unclear what teaching, if any, of *Chefalas* that the Examiner relies upon as providing the recited VDL file.

Appellant notes that *Hill* does provide a Network Status Display 42, *see e.g.* the abstract of *Hill*. However, it is unclear whether the Final Office Action is relying upon this teaching of *Hill* as providing a VDL file.

Irrespective of the above-noted deficiency in the Final Office Action, neither *Chefalas* nor *Hill* teaches or suggests a VDL file as recited by independent claim 1, as discussed further hereafter.

First, *Chefalas* fails to teach or suggest a VDL file as recited by claim 1 that specifies those items recited in claim 1. *Chefalas* discloses "a business service for automatic detection, notification and elimination of viruses for a large network of machines." Para. 0012 of *Chefalas*. *Chefalas* explains its system (in para. 0012) as follows:

> A software subsystem known as a virus scanner and notifier (VSN), residing on a client data processing system monitors for viruses. In response to detecting a virus infection, the VSN at the client data processing system sends notification of a presence of the virus on the data processing system to a software module known as the virus scanner controller (VSC) residing at a server, wherein the notification includes an identification of an action taken in response to detecting the virus. Further, the VSN at the client data processing

> system may take actions to eliminate or quarantine the virus. In a server data processing system, a notification of a presence of a virus on a client data processing system is received through a communications link. The communication with the client data processing system through the communications link is severed in response to receiving the notification. Virus removal processes may be executed on the server data processing system. Alternatively or additionally, the VSC module at the server data processing system may execute an action based on a business policy in response to receiving the notification.

Chefalas further describes that information may be communicated between a VSN on a client and a VSC on a server in the form of "business events". Examples of such business events are shown in FIGS. 4A and 4B of *Chefalas*. "In FIG. 4A, business event 400 may be an event sent from a VSN at the client to a VSC at the server, providing notification of an action taken on the client." Para. 0043 of *Chefalas*. "In FIG. 4B, business event 412 is an example of a business event sent from a server to a client or from one server to another server." Para. 0045 of *Chefalas*. As shown in FIG. 4A, the business event 400 may include information specifying the name of a virus detected on a client, the action taken on the client, the computer ID of the client, as well as a header. As shown in FIG. 4B, the business event 412 may include instructions from the server to the client, as well as a header.

Chefalas does not teach or suggest that its business events constitute a human-readable and machine-readable VDL file as recited in claim 1. First, the business events of *Chefalas* are not described as being both human-readable and machine-readable. Rather, the business events of *Chefalas* are merely notification "messages" used to exchange information/instructions between a server and a client in response to a virus being detected on the client. It appears that the business event messages are merely machine-readable messages that can be processed by a VSC process on a server or by a VSN process on a client. *Chefalas* teaches that the business events can be used by the processes to enable the system to automatically respond to a detected virus without requiring any manual intervention, *see e.g.*, para. 0012 of *Chefalas*. Thus, there is no suggestion by *Chefalas* that the business events are human-readable, as they are used solely for processing by computer processes, rather than by humans.

Further, the business event messages of *Chefalas* do not specify at least one policy definition with respect to detecting the vulnerability of the computer system to the specified attack, which the Final Office Action appears to concede on page 3 thereof.

Hill also fails to teach or suggest a VDL file as recited by independent claim 1. The Final Office Action appears to assert that the Network Status Display 42 of *Hill* provides such a VDL file, *see* page 2 of the Final Office Action. Appellant disagrees. The Network Status Display 42 of *Hill* is not a file at all, but is instead a display on which information may be displayed.

Further, to the extent that the Final Office Action is relying upon Network Status Display 42 (or any other portion of *Hill* for that matter) as teaching or suggesting the recited VDL file, the Final Office Action fails to establish a prima facie case of obviousness. As set forth in M.P.E.P. § 2141, "[o]ffice policy is to follow *Graham v. John Deere Co.* in the consideration and determination of obviousness under 35 U.S.C. 103". The Final Office Action fails to provide the appropriate showings necessary for establishing a prima facie case of obviousness to the extent that Network Status Display 42 or any other portion of *Hill* is relied upon as teaching or suggesting a VDL file. For instance, the Final Office Action fails to identify the lack of such teaching by *Chefalas*, and the Final Office Action fails to provide any motivation or any likelihood of success in combining the Network Status Display 42 of *Hill* with *Chefalas*. As mentioned above, *Chefalas* teaches a system in which no manual intervention is required, and thus a Network Status Display 42 as taught by *Hill* appears unnecessary in the *Chefalas* system.

Thus, for at least the above reasons, the rejection of claim 1 should be overturned.

Claims 2 and 6-11 each depend either directly or indirectly from independent claim 1, and are thus likewise believed to be allowable at least based on their dependency from claim 1 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 2 and 6-11 also be overturned.

Dependent Claim 3

Dependent claim 3 depends from independent claim 1 and, thus, includes all of the limitations of independent claim 1 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 3 is allowable at least because of its dependence from claim 1 for the reasons discussed above.

Further, claim 3 recites "generating the VDL file specifying a computing platform of the computer system." The combination of *Chefalas* and *Hill* fails to teach of suggest such a VDL file that specifies a computing platform of a computer system. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0047 of *Chefalas*, *see* page 3 of the Final Office Action. Paragraph 0047 of *Chefalas* merely provides:

> In FIG. 5B, policy 502 identifies actions based on the identification of the client based on the computer ID. In entry 516 computer A is disconnected and the action taken at computer A is logged if the business event identifies the virus as being detected at computer A. If the business event originates from computer B, router C is disabled and the action taken at computer B is logged as illustrated in entry 518. If the business event is identified as originated from computer C, the action taken is to page a technician, email a manager, and log the action taken at computer C as shown in entry 520.

The above teaching of *Chefalas* does not teach or suggest a VDL file that is machine-readable and human-readable. For instance, *Chefalas* does not teach or suggest that policy 502 is both machine-readable and human-readable. Further, the above teaching of *Chefalas* does not teach or suggest specifying a computer platform of a computer system. Rather, it merely mentions that a computer ID may identify a computer on which a virus is detected. Such a computer ID is not taught or suggested as specifying the platform of the computer, but instead merely identifies the computer. Thus, *Chefalas* fails to teach or suggest this further element of claim 3.

Additionally, *Hill* is not relied upon as teaching this element of claim 3, nor does it do so. Thus, for at least the above reasons, the rejection of claim 3 should be overturned.

Dependent Claim 4

Dependent claim 4 depends from independent claim 1 and, thus, includes all of the limitations of independent claim 1 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 4 is allowable at least because of its dependence from claim 1 for the reasons discussed above.

Further, claim 4 recites "generating the VDL file: specifying a security category of the specified attack; and specifying at least one policy group with respect to the specified security category." The combination of *Chefalas* and *Hill* fails to teach of suggest such a VDL file that is both machine-readable and human-readable (per claim 1) and that specifies this further information. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0046 of *Chefalas*, *see* page 3 of the Final Office Action. Paragraph 0046 of *Chefalas* merely provides:

> Turning now to FIGS. 5A and 5B, illustrations of policies for taking action in response to notification of a virus are depicted in accordance with a preferred embodiment of the present invention. Policy 500 in FIG. 5A and policy 502 in FIG. 5B are examples of rules that may be used to implement business decisions as to how to handle the notification of the presence of a virus within a network data processing system. In the depicted examples, policy 500 provides for different actions based on the name of the virus, as illustrated in entries 504-514. The virus names are used as indexes into policy 500. For example, if virus A is present, entry 504 merely logs the action taken at the client. An occurrence of virus B or virus C results in the scheduling of maintenance of the client and logging of the client as shown in entries 506 and 508. The presence of virus D indexes to entry 510, which results in a manager being paged, the client and shared resources being disconnected, and the action taken at the client being logged. The occurrence of virus F results in a technician being paged and the client being disconnected as shown in entry 514.

The above teaching of *Chefalas* does not teach or suggest a VDL file that is machine-readable and human-readable. For instance, *Chefalas* does not teach or suggest that policies 500 and 502 are both machine-readable and human-readable. Further, the above teaching of *Chefalas* does not teach or suggest specifying a security category of a specified attack. Thus, *Chefalas* fails to teach or suggest this further element of claim 4.

Additionally, *Hill* is not relied upon as teaching this element of claim 4, nor does it do so. Thus, for at least the above reasons, the rejection of claim 4 should be overturned.

Dependent Claim 5

Dependent claim 5 depends from independent claim 1 and, thus, includes all of the limitations of independent claim 1 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 5 is allowable at least because of its dependence from claim 1 for the reasons discussed above.

Further, claim 5 recites "generating the VDL file specifying a vulnerability scanner executing on the computer system." The combination of *Chefalas* and *Hill* fails to teach of suggest such a VDL file that specifies a vulnerability scanner executing on the computer system. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraphs 0027-0028 of *Chefalas*, *see* page 4 of the Final Office Action. Paragraphs 0027-0028 of *Chefalas* merely provide:

> A business service is a business model in which a software application is deployed to a customer as a service on a subscription-fee basis. Customers subscribe to the service and the service provider charges its customers a monthly rate, fixed or variable, for providing the service. The service provider is responsible for the equipment and infrastructure needed to provide and deliver the service. The service provider also maintains the service by providing periodic software updates, functional enhancements, and support for the service. Server 106 at the customer premises has a virus scanner and notifier module within VSC 126 to coordinate activity and receive events from the virus scanner and notifier module located at clients 110-118 on the network. Although a single server is illustrated, the mechanism of the present invention may be implemented using multiple servers.
>
> If a virus is detected on a client, such as client 112, software agent, VSN 128, installed on the client 112 immediately quarantines the offending file and notifies VSC 126 at server 106 via network 104 that a virus has been detected. If the detected virus is the type of virus that can be replicated or cloned, VSC 126 at server 106 immediately severs the connection with client 112 and all other clients connected to the server. Further, VSC 126 at server 106 initiates the virus removal processes on clients 110-118. Server 106 also removes any network shares under its control. Then, VSC 126 at server 106 runs the anti-virus software on the server, removing and quarantine any infected files. Server 106 may then decide to shut down to protect itself and the network shares it controls.

The above teaching of *Chefalas* merely mentions that a virus scanner may be implemented at clients. *Chefalas* provide no teaching or suggestion of a VDL file that is machine-readable and human-readable (per claim 1). Further, *Chefalas* provides no teaching of such a VDL file specifying a vulnerability scanner. Even if the above-mentioned virus scanner is considered a vulnerability scanner (without conceding this point), *Chefalas* provides no teaching of specifying such a virus scanner in a VDL file. Thus, *Chefalas* fails to teach or suggest this further element of claim 5.

Additionally, *Hill* is not relied upon as teaching this element of claim 5, nor does it do so. Thus, for at least the above reasons, the rejection of claim 5 should be overturned.

Independent Claim 12 and Dependent Claims 16-19

Independent claim 12 recites:

> A method of defining a security vulnerability condition of a system, comprising:
> generating a human-readable and machine-readable vulnerability description language (VDL) file specifying:
> a name of a vulnerability associated with the system;
> at least one attribute of the specified vulnerability;
> a remedy for the vulnerability according to the specified computing platform;
> a policy definition with respect to detecting the specified vulnerability; and
> at least one attribute of the specified policy definition.

The combination of *Chefalas* and *Hill* fails to teach or suggest all elements of claim 12. The Final Office Action provides the same explanation in rejecting both claims 1 and 12, *see* pages 2-3 of the Final Office Action. As discussed above with claim 1, the Final Office Action fails to establish that the combination of *Chefalas* and *Hill* teaches or suggests generating a human-readable and machine-readable VDL file, as recited by claim 12. Again, *Chefalas* fails to teach or suggest such a VDL file that is both human-readable and machine-readable. Further, *Hill* is not relied upon as teaching this element, nor does it do so. To the extent that the Final Office Action relies upon Network Status Display 42 of *Hill* as teaching or suggesting this element, Appellant disagrees. First, the Network Status Display 42 is not a file, but is instead a display on which information may be output. Further, the Final Office

Action fails to provide the necessary showing under 35 U.S.C. §103 for establishing a combination of Network Status Display 42 with the system of *Chefalas*.

Thus, for at least the above reasons, the rejection of claim 12 should be overturned.

Claims 16-19 each depend either directly or indirectly from independent claim 12, and are thus likewise believed to be allowable at least based on their dependency from claim 12 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 16-19 also be overturned.

Dependent Claim 13

Dependent claim 13 depends from independent claim 12 and, thus, includes all of the limitations of independent claim 12 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 13 is allowable at least because of its dependence from claim 12 for the reasons discussed above.

Further, claim 13 recites "specifying a computing platform of the system." The combination of *Chefalas* and *Hill* fails to teach of suggest specifying a computing platform of a system. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0047 of *Chefalas*, *see* page 3 of the Final Office Action. Paragraph 0047 of *Chefalas* merely provides:

> In FIG. 5B, policy 502 identifies actions based on the identification of the client based on the computer ID. In entry 516 computer A is disconnected and the action taken at computer A is logged if the business event identifies the virus as being detected at computer A. If the business event originates from computer B, router C is disabled and the action taken at computer B is logged as illustrated in entry 518. If the business event is identified as originated from computer C, the action taken is to page a technician, email a manager, and log the action taken at computer C as shown in entry 520.

The above teaching of *Chefalas* does not teach or suggest specifying a platform of a system. Rather, it merely mentions that a computer ID may identify a computer on which a virus is detected. Such a computer ID is not taught or suggested as specifying the platform of the computer, but instead merely identifies the computer. Thus, *Chefalas* fails to teach or suggest this further element of claim 13.

Additionally, *Hill* is not relied upon as teaching this element of claim 13, nor does it do so. Thus, for at least the above reasons, the rejection of claim 13 should be overturned.

Dependent Claim 14

Dependent claim 14 depends from independent claim 12 and, thus, includes all of the limitations of independent claim 12 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 14 is allowable at least because of its dependence from claim 12 for the reasons discussed above.

Further, claim 14 recites "specifying a security category of the specified vulnerability; and specifying at least one policy group with respect to the specified security category." The combination of *Chefalas* and *Hill* fails to teach of suggest specifying this further information. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0046 of *Chefalas*, *see* page 3 of the Final Office Action. Paragraph 0046 of *Chefalas* merely provides:

> Turning now to FIGS. 5A and 5B, illustrations of policies for taking action in response to notification of a virus are depicted in accordance with a preferred embodiment of the present invention. Policy 500 in FIG. 5A and policy 502 in FIG. 5B are examples of rules that may be used to implement business decisions as to how to handle the notification of the presence of a virus within a network data processing system. In the depicted examples, policy 500 provides for different actions based on the name of the virus, as illustrated in entries 504-514. The virus names are used as indexes into policy 500. For example, if virus A is present, entry 504 merely logs the action taken at the client. An occurrence of virus B or virus C results in the scheduling of maintenance of the client and logging of the client as shown in entries 506 and 508. The presence of virus D indexes to entry 510, which results in a manager being paged, the client and shared resources being disconnected, and the action taken at the client being logged. The occurrence of virus F results in a technician being paged and the client being disconnected as shown in entry 514.

The above teaching of *Chefalas* does not teach or suggest specifying a security category of a specified vulnerability. Thus, *Chefalas* fails to teach or suggest this further element of claim 14.

Additionally, *Hill* is not relied upon as teaching this element of claim 14, nor does it do so. Thus, for at least the above reasons, the rejection of claim 14 should be overturned.

Dependent Claim 15

Dependent claim 15 depends from independent claim 12 and, thus, includes all of the limitations of independent claim 12 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 15 is allowable at least because of its dependence from claim 12 for the reasons discussed above.

Further, claim 15 recites "specifying a vulnerability scanner executing on the system." The combination of *Chefalas* and *Hill* fails to teach of suggest specifying a vulnerability scanner executing on the system. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraphs 0027-0028 of *Chefalas*, *see* page 4 of the Final Office Action. Paragraphs 0027-0028 of *Chefalas* merely provide:

> A business service is a business model in which a software application is deployed to a customer as a service on a subscription-fee basis. Customers subscribe to the service and the service provider charges its customers a monthly rate, fixed or variable, for providing the service. The service provider is responsible for the equipment and infrastructure needed to provide and deliver the service. The service provider also maintains the service by providing periodic software updates, functional enhancements, and support for the service. Server 106 at the customer premises has a virus scanner and notifier module within VSC 126 to coordinate activity and receive events from the virus scanner and notifier module located at clients 110-118 on the network. Although a single server is illustrated, the mechanism of the present invention may be implemented using multiple servers.
>
> If a virus is detected on a client, such as client 112, software agent, VSN 128, installed on the client 112 immediately quarantines the offending file and notifies VSC 126 at server 106 via network 104 that a virus has been detected. If the detected virus is the type of virus that can be replicated or cloned, VSC 126 at server 106 immediately severs the connection with client 112 and all other clients connected to the server. Further, VSC 126 at server 106 initiates the virus removal processes on clients 110-118. Server 106 also removes any network shares under its control. Then, VSC 126 at server 106 runs the anti-virus software on the server, removing and quarantine any infected files. Server 106 may then decide to shut down to protect itself and the network shares it controls.

The above teaching of *Chefalas* merely mentions that a virus scanner may be implemented at clients. *Chefalas* provides no teaching of specifying a vulnerability scanner. Even if the above-mentioned virus scanner is considered a vulnerability scanner (without conceding this point), *Chefalas* provides no teaching of specifying such a virus scanner, but

instead *Chefalas* merely mentions that a virus scanner may be located at clients. Thus, *Chefalas* fails to teach or suggest this further element of claim 15.

Additionally, *Hill* is not relied upon as teaching this element of claim 15, nor does it do so. Thus, for at least the above reasons, the rejection of claim 15 should be overturned.

Independent Claim 20 and Dependent Claims 21 and 24-27

Independent claim 20 recites:

> A system of defining security vulnerabilities of a computer system, comprising:
> a human-readable and machine-readable vulnerability description language (VDL) file containing a definition of at least one vulnerability and a definition of at least one policy item for detecting the vulnerability;
> an interpreter operable to parse the at least one vulnerability definition and at least one policy item definition in the VDL file and organize the parsed definitions pursuant to a predetermined format; and
> a data storage operable to store the parsed and organized at least one vulnerability and at least one policy item definition, wherein the data storage is accessible by at least one vulnerability scanner application.

The combination of *Chefalas* and *Hill* fails to teach or suggest all elements of claim 20. The Final Office Action asserts that *Chefalas* teaches a human-readable and machine-readable VDL file as recited by claim 20. As discussed above with claim 1, *Chefalas* fails to teach or suggest such a human-readable and machine-readable VDL file. For instance, the business events of *Chefalas* are merely notification "messages" used to exchange information/instructions between a server and a client in response to a virus being detected on the client. There is no suggestion by *Chefalas* that the business events are human-readable, as they are used solely for processing by computer processes, rather than by humans. *Hill* is not relied upon as teaching a human-readable and machine-readable VDL file, nor does it do so.

Further, the Final Office Action asserts that *Chefalas* teaches the recited interpreter of claim 20. Appellant disagrees. For instance, *Chefalas* provides no teaching or suggestion of an interpreter that parses at least one vulnerability definition and at least one policy item definition in the VDL file and organizes the parsed definitions pursuant to a predetermined format. While *Chefalas* provides examples of the formats of the business events themselves

(e.g., in FIGS. 4A-4B), *Chefalas* does not mention an interpreter that receives a business event and parses information contained therein to organize the parsed information into a predetermined format. *Hill* is not relied upon as teaching or suggesting such an interpreter, nor does it do so.

Thus, the combination of *Chefalas* and *Hill* fails to teach or suggest at least the above elements of claim 20. Therefore, the rejection of claim 20 should be overturned.

Claims 21 and 24-27 each depend either directly or indirectly from independent claim 20, and are thus likewise believed to be allowable at least based on their dependency from claim 20 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 21 and 24-27 also be overturned.

Dependent Claim 22

Dependent claim 22 depends from independent claim 20 and, thus, includes all of the limitations of independent claim 20 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 22 is allowable at least because of its dependence from claim 20 for the reasons discussed above.

Further, claim 22 recites "wherein the VDL file further comprises a definition of a vulnerability scanner application." The combination of *Chefalas* and *Hill* fails to teach of suggest such a VDL file that comprises a definition of a vulnerability scanner application. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0045 of *Chefalas*, *see* page 7 of the Final Office Action. Paragraph 0045 of *Chefalas* merely provides:

> In FIG. 4B, business event 412 is an example of a business event sent from a server to a client or from one server to another server. Business event 412 takes the form of a data packet having a header 414 and a payload 416. In this example, payload 416 contains an instruction 418. If sent to a client from a server, the instruction may be, for example, to initiate a virus checking process. If sent from one server to another server, the instruction may be, for example, to shut down the server receiving business event 412.

The above teaching of *Chefalas* merely mentions that a business event may contain an instruction to initiate a virus checking process. *Chefalas* provides no teaching or suggestion

of a VDL file that is machine-readable and human-readable (per claim 20). Further, *Chefalas* provides no teaching of such a VDL file comprising a definition of a vulnerability scanner application. Even if the above-mentioned virus checking process is considered a vulnerability scanner (without conceding this point), *Chefalas* provides no teaching of defining such a virus checking process in a VDL file that is machine-readable and human-readable. Thus, *Chefalas* fails to teach or suggest this further element of claim 22.

Additionally, *Hill* is not relied upon as teaching this element of claim 22, nor does it do so. Thus, for at least the above reasons, the rejection of claim 22 should be overturned.

Dependent Claim 23

Dependent claim 23 depends from independent claim 20 and, thus, includes all of the limitations of independent claim 20 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 23 is allowable at least because of its dependence from claim 20 for the reasons discussed above.

Further, claim 23 recites "wherein the VDL file further comprises a definition of a security category providing a grouping of the at least one vulnerability, and a definition of a policy group providing a grouping of the at least one policy item." The combination of *Chefalas* and *Hill* fails to teach of suggest such a VDL file that is both machine-readable and human-readable (per claim 20) and that specifies this further information. The Final Office Action asserts that *Chefalas* teaches this element, citing to paragraph 0046 of *Chefalas*, *see* page 7 of the Final Office Action. Paragraph 0046 of *Chefalas* merely provides:

> Turning now to FIGS. 5A and 5B, illustrations of policies for taking action in response to notification of a virus are depicted in accordance with a preferred embodiment of the present invention. Policy 500 in FIG. 5A and policy 502 in FIG. 5B are examples of rules that may be used to implement business decisions as to how to handle the notification of the presence of a virus within a network data processing system. In the depicted examples, policy 500 provides for different actions based on the name of the virus, as illustrated in entries 504-514. The virus names are used as indexes into policy 500. For example, if virus A is present, entry 504 merely logs the action taken at the client. An occurrence of virus B or virus C results in the scheduling of maintenance of the client and logging of the client as shown in entries 506 and 508. The presence of virus D indexes to entry 510, which results in a manager being paged, the client and shared resources being disconnected, and the

> action taken at the client being logged. The occurrence of virus F results in a technician being paged and the client being disconnected as shown in entry 514.

The above teaching of *Chefalas* does not teach or suggest a VDL file that is machine-readable and human-readable. For instance, *Chefalas* does not teach or suggest that policies 500 and 502 are both machine-readable and human-readable. Further, the above teaching of *Chefalas* does not teach or suggest a definition of a security category providing a grouping of the at least one vulnerability. Thus, *Chefalas* fails to teach or suggest this further element of claim 23.

Additionally, *Hill* is not relied upon as teaching this element of claim 23, nor does it do so. Thus, for at least the above reasons, the rejection of claim 23 should be overturned.

B. Conclusion

In view of the above, Appellant requests that the board overturn the outstanding rejections of claims 1-27. Attached hereto are a Claims Appendix, Evidence Appendix, and Related Proceedings Appendix. As noted in the attached Evidence Appendix, no evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted. Also, as noted by the Related Proceedings Appendix, no related proceedings are referenced in II above, and thus no copies of decisions in related proceedings are provided.

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV568259959US in an envelope addressed to: MS Appeal Brief - Patents, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: April 20, 2006

Typed Name: Elise Perkins

Signature: Elise Perkins

Respectfully submitted,

By: _____

Jody C. Bishop
Attorney/Agent for Applicant(s)
Reg. No. 44,034
Date: April 20, 2006
Telephone No. (214) 855-8007

VIII. CLAIMS APPENDIX

Claims Involved in the Appeal of Application Serial No. 10/001,410

1. A method of defining the security vulnerability of a computer system, comprising:

generating a human-readable and machine-readable vulnerability description language (VDL) file specifying:

an attack representing a recognized vulnerability of the computer system;

at least one attribute of the specified attack;

at least one policy definition with respect to detecting the vulnerability of the computer system to the specified attack; and

a remedy for the specified vulnerability.

2. The method, as set forth in claim 1 further comprising generating the VDL file specifying at least one attribute of the specified policy definition.

3. The method, as set forth in claim 1 further comprising generating the VDL file specifying a computing platform of the computer system.

4. The method, as set forth in claim 1 further comprising generating the VDL file:

specifying a security category of the specified attack; and

specifying at least one policy group with respect to the specified security category.

5. The method, as set forth in claim 1 further comprising generating the VDL file specifying a vulnerability scanner executing on the computer system.

6. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying an identification of the severity associated with a breach of the computer system by the attack.

7. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying a description of the attack.

8. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying an explanation of why the specified attack is important.

9. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying how information is to be reported to a user with respect to the specified attack.

10. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying a source of a remedy operable to fix the specified vulnerability.

11. The method, as set forth in claim 1 wherein specifying at least one attribute of the specified attack comprises specifying information to enable a manual remedy of the specified vulnerability.

12. A method of defining a security vulnerability condition of a system, comprising:

generating a human-readable and machine-readable vulnerability description language (VDL) file specifying:

a name of a vulnerability associated with the system;

at least one attribute of the specified vulnerability;

a remedy for the vulnerability according to the specified computing platform;

a policy definition with respect to detecting the specified vulnerability; and

at least one attribute of the specified policy definition.

13. The method, as set forth in claim 12 further comprising specifying a computing platform of the system.

14. The method, as set forth in claim 12 further comprising:

specifying a security category of the specified vulnerability; and

specifying at least one policy group with respect to the specified security category.

15. The method, as set forth in claim 12 further comprising specifying a vulnerability scanner executing on the system.

16. The method, as set forth in claim 12 wherein specifying at least one attribute of the specified vulnerability comprises specifying an identification of the severity associated with a breach of the specified vulnerability.

17. The method, as set forth in claim 12 wherein specifying at least one attribute of the specified vulnerability comprises specifying an explanation of why the specified vulnerability is important.

18. The method, as set forth in claim 12 wherein specifying at least one attribute of the specified vulnerability comprises specifying how information is to be reported to a user in response to detecting the specified vulnerability.

19. The method, as set forth in claim 12 wherein specifying at least one attribute of the specified vulnerability comprises specifying an application operable to respond to a detection of the specified vulnerability.

20. A system of defining security vulnerabilities of a computer system, comprising:

a human-readable and machine-readable vulnerability description language (VDL) file containing a definition of at least one vulnerability and a definition of at least one policy item for detecting the vulnerability;

an interpreter operable to parse the at least one vulnerability definition and at least one policy item definition in the VDL file and organize the parsed definitions pursuant to a predetermined format; and

a data storage operable to store the parsed and organized at least one vulnerability and at least one policy item definition, wherein the data storage is accessible by at least one vulnerability scanner application.

21. The system, as set forth in claim 20 wherein the data storage is a relational database having a plurality of tables.

22. The system, as set forth in claim 20 wherein the VDL file further comprises a definition of a vulnerability scanner application.

23. The system, as set forth in claim 20 wherein the VDL file further comprises a definition of a security category providing a grouping of the at least one vulnerability, and a definition of a policy group providing a grouping of the at least one policy item.

24. The system, as set forth in claim 20 wherein the VDL file further comprises a definition of at least one attribute of the at least one vulnerability.

25. The system, as set forth in claim 20 wherein the VDL file further comprises an identification of the severity of risk associated with the at least one vulnerability.

26. The system, as set forth in claim 20 wherein the VDL file further comprises a definition of how information is to be displayed to a user with respect to the at least one vulnerability.

27. The system, as set forth in claim 20 wherein the VDL file further comprises a definition of an application operable to respond to detecting the at least one vulnerability.

IX. EVIDENCE APPENDIX

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings are referenced in II above, and thus no copies of decisions in related proceedings are provided.